(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 0 798 673 A1

(12) EUROPEAN PATENT APPLICATION

(43) Date of publication:
01.10.1997 Bulletin 1997/40

(51) Int. Cl.⁶: G07F 7/10

(21) Application number: 96200867.8

(22) Date of filing: 29.03.1996

(84) Designated Contracting States:
AT BE CH DE DK ES FI FR GB GR IE IT LI LU MC NL PT SE

(71) Applicant: Koninklijke PTT Nederland N.V.
2509 CH Den Haag (NL)

(72) Inventor: Drupsteen, Michiel Marco Paul
2264 XZ Leidschendam (NL)

(74) Representative: Beitsma, Gerhard Romano
Koninklijke PTT Nederland N.V.,
P.O. Box 95321
2509 CH Den Haag (NL)

(54) Method of securely loading commands in a smart card

(57) The invention relates to a method of securely loading and validating commands (COM) in a smart card (SC). Especially in the case where application-specific commands are loaded by an application provider (AP), that is off-line with respect to the card issuer (CI), it must be ensured that the commands are valid. The invention provides a method involving the protection of the commands (COM) by means of authentication codes, these codes (MAC1, MAC2) being produced using two different keys: one key (K1) is stored by the card issuer (CI), the other (K2) by a trusted third party (TTP). A further authentication code (MAC3), produced using a key from a set of keys (K3*), may be utilized to selectively validate commands for individual applications (e.g. AP1, AP2).

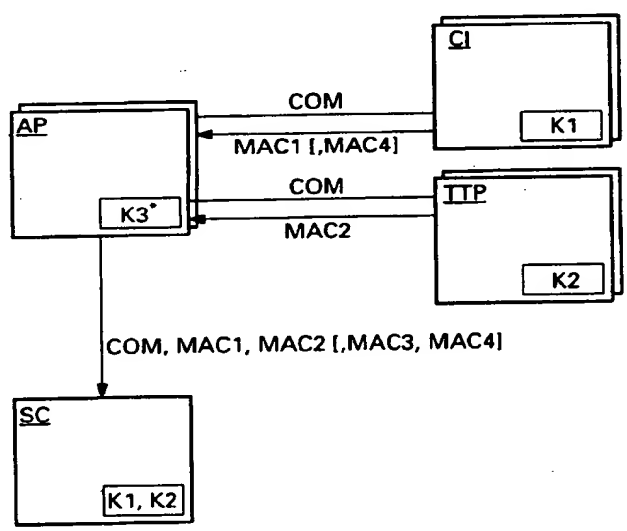

Fig. 3

EP 0 798 673 A1

Description

BACKGROUND OF THE INVENTION

The present invention relates to a method of loading commands in a smart card. More specifically, the present invention relates to a method of off-line loading application specific commands in a smart card.

In modern payment systems, the use of electronic payment means becomes increasingly important. Electronic payment means, such as memory cards and smart cards, are gaining acceptance as their applications are expanded. In many contries electronic cards are being used for public telephones and the like. Advanced cards are capable of containing electronic "purses", in addition to other functionalities. Such advanced payment means contain, in addition to a memory, a processor capable of running suitable programs.

It should be noted that in this text, the terms smart card or card will be used to denote electronic payment means having at least one integrated electronic circuit comprising a processor and a memory. The actual shape of a so-called smart card is not of importance.

The programs running on the processor of a smart card determine the services offered by the card, that is, the functions and associated data structures (e.g. purse, user identification, loyalty program) of the smart card depend on the software present in the card. As time passes, the need often arises to update the programs of the card, for example in order to add a new function or to improve an existing function. To this end, the card should be able to accept new programs which may replace other programs. However, it must be ascertained that the newly loaded programs are valid.

Authentication of programs can relatively easily be accomplished by using a secure data exchange protocol between the card issuer and the card. However, other parties, such as applications providers, want to be able to load new applications (and hence new commands) into a card without having to do this via the card issuer.

SUMMARY OF THE INVENTION

It is an object of the invention to overcome the above-mentioned and other disadvantages and to provide a method which allows commands to be loaded and activated in a smart card in a secure manner. It is a further object of the present invention to provide a method which allows application-specific commands to be loaded in a smart card by an application provider while being able to guarantee the integrity of the commands.

These and other objects are achieved by a method of securely loading commands in a smart card by a first party, the card being issued by a second party, the method according to the invention comprising the steps of:

- the second party producing a first authentication code of a command using a first key,
- a third party producing a second authentication code of the command using a second key,
- transferring the command with the codes to the card,
- the card validating the command by reproducing the first and second authentication codes using the first and the second key respectively and comparing the reproduced codes with the transferred codes.

In the method of the present invention, the authenticity of the commands is thus safeguarded by having two different and preferably independent parties produce an authentication code: the second party, which normally is the card issuer, and a third party, which normally is a trusted independent party, such as a central bank. Once both authentication codes have been produced, it is virtually impossible to alter the commands without this being noticeable by means of the authentication codes. The first party, the application provider, has proof that both the first and second parties have "certified" the command. Preferably the third party stores copies of the command.

It will be understood that the method of the invention applies to both the loading of a single command and the loading of a set of commands.

The transferring to the card and the subsequent validation are preferably repeated when the validation fails, the use of the loaded commands being blocked until the validation succeeds. This prevents invalid commands, i.e. commands affected by transmission errors or manipulations, to be executed by the card.

Advantageously, the loaded commands are permanently disabled when the validation fails a predetermined number of times, the number preferably being less than ten. The disabling of the loaded commands may be performed by changing the key of the second party. In this way, the indefinite reloading of corrupted commands can be terminated.

The commands loaded may be application-specific commands, i.e. commands directly influencing the applications of the card. Such commands are often machine language instructions which in principle allow manipulation of the card applications. By using the method of the present invention, the use of such commands is controlled whereby the integrity of the applications is ensured.

The first and/or second authentication codes are preferably message authentication codes produced according to the ANSI X9.19 standard.

Advantageously, an additional authentication code is produced by the second party, said additional code not involving the first or second key. This allows the first party a validity check without influencing the method as such.

BRIEF DESCRIPTION OF THE DRAWINGS

Fig. 1 shows a smart card as may be used in the method of the present invention.

Fig. 2 schematically shows the integrated circuit of the smart card of Fig. 1.

Fig. 3 schematically shows the exchange of data in the method of the present invention.

Fig. 4 schematically shows data structures on a smart card.

Fig. 5 schematically shows a flag register as used in the method of the present invention.

Fig. 6 schematically shows an exemplary embodiment of the method of the present invention.

EXEMPLARY EMBODIMENTS

The smart card or IC card 1 shown by way of example in Fig. 1 comprises a substrate 2, in which an integrated circuit is embedded. The integrated circuit is provided with contacts 3 for contacting a card reader or the like. It should be noted that the present invention can also be applied in the case of so-called contactless smart cards.

The integrated circuit 10 shown schematically and by way of example in Fig. 2 comprises a processor 11, a memory 12 and an input/output circuit 13. The memory may comprise a volatile (RAM) memory part for temporarily storing data and a non-volatile (ROM) memory part for permanently or semi-permanently storing data. The latter part is preferably an EEPROM type memory. The data stored in the non-volatile part may contain both programming data (instructions, programs) and payment data, i.e. data relating to monetary transactions. It will be understood that a separate memory (not shown) may be provided to store the instruction set of the processor 11. The input-output circuit 13 may communicate with external devices via contacts (not shown in Fig. 2), such as the contacts 3 shown in Fig. 1. The integrated circuit 10 may be contained in the card 1 of Fig. 1.

An embodiment of the method according to the invention is shown schematically and by way of example in Fig. 3. A first party, e.g. an application provider AP, offers card functions to its customers. A smart card SC is issued by a second party, a card issuer CI. According to the invention, a trusted third party TTP is involved in the validation of commands, as will be explained later.

A set of commands COM may have been produced by the card issuer CI or by an external source by order of the applications provider AP. It should be noted that the set of commands COM may comprise one or more commands. That is, the method as described here may be applied to individual commands or to a set of commands. In the following description, a single command rather than a set will be used as an example to explain the method of the invention. The commands may be application-specific commands (ASC) or general purpose commands (GPC).

As shown in Fig. 3, the application provider AP offers a command (COM) to the card issuer CI for authentication. A first authentication code MAC1, based on the command COM, is produced by the card issuer CI using a first key K1. In addition to the code MAC1 the set of commands may optionally receive an additional (fourth) authentication code MAC4 which does not involve the use of a key, at least not the key K1. The additional authentication code MAC4 serves to provide an additional verification of the command, independent of the keys used in the other authentication codes. The computation of the additional authentication code MAC4 may involve the first authentication code MAC1. In that case, the additional authentication code can be written as:

$$MAC4 = F_{K1}(COM, MAC1),$$

where F denotes a function by which the code is determined. The authentication code MAC1, and optionally the additional authentication code MAC4, is transferred to the application provider AP, where it is temporarily stored.

According to the present invention, the command COM is also offered to a trusted third party TTP. The third party produces a second authentication code MAC2 of the command COM, using a second key K2. The second authentication code MAC2 is transferred to the application provider AP. The third party is e.g. a central bank or an institution appointed by the card issuer and the service provider.

The application provider AP may produce a third authentication code MAC3 using a third key K3*. It should be noted that the key K3* may comprise a set of keys (as denoted by the asterisk), each individual key corresponding with an individual card application and/or individual card file. The key K3*, which may thus consist of a set of keys K3-1, K3-2, etc., may be used to selectively load commands into specific applications and/or files of the card. This will further be explained with reference to Figs. 4 and 5. It should be noted that as the third key (K3) may vary between applications, the third authentication code (MAC3) also varies. That is, the third authentication code (MAC3) may constitute an application-specific authentication code. It should be noted that the third authentication code may be based on both the command COM and the first and second authentication codes, thus providing a double authentication. In that case, the third authentication code can be written as:

$$MAC3 = F_{K3}(COM, MAC1, MAC2),$$

where F denotes a function by which the code is determined.

The authentication codes MAC1 and MAC2, as well as the authentication codes MAC3 and MAC4, are preferably produced using a message authentication scheme involving encryption. Such a scheme is e.g. disclosed in the ANSI X9.19 standard. It will be understood that the actual scheme (or schemes) used in producing the message authentication codes is not essential to the present invention. The additional authentication code

may also be produced using encryption (e.g. using a public key), or may be produced using a so-called hash function.

Both authentication codes MAC1 and MAC2, as well as the optional authentication codes MAC3 and MAC4, may be appended to the command, or be associated with the command in another manner. In Fig. 3, the command and its associated authentication codes is denoted by:

COM, MAC1, MAC2 [,MAC3, MAC4] ,

the square brackets indicating the optional codes. The actual order in which the command and its associated codes is transferred may of course vary.

Upon receipt of the command by the card SC, the card reproduces the authentication codes MAC1 and MAC2 using the keys K1 and K2 which have been stored in the card previously. The reproduced codes MAC1' and MAC2' are compared with the received codes MAC1 and MAC2. If the reproduced and received codes are identical, the validation of the commands has succeeded and the commands may be activated. This activation may be done by setting (or resetting) a certain flag. If the validation fails, i.e. if the corresponding received and reproduced codes are not identical, a request for retransmission of the commands may be issued. A retransmission counter keeps track of the number of retransmissions and inhibits retransmission if a predetermined number (e.g. 5) is exceeded.

The card is thus preferably arranged in such a way that the commands loaded into the card cannot be used until they are activated. That is, the loaded commands are blocked until they are released as a result of a positive validation. This prevents invalid commands to be executed by the card.

Fig. 4 schematically shows data structures on a smart card, such as the card 1 of Fig. 1, according to a preferred embodiment of the present invention. The data structures are in practice stored in a memory, such as the memory 12 of Fig. 2.

A so-called master file MF contains, i.a., references to other files. Other files are associated with individual applications. In Fig. 4, several so-called application files (AP1, AP2, AP3) are shown. Each application (or application file) AP1, AP2, ..., may comprise one or more subfiles, e.g. program files and data files.

As shown in Fig. 4, each application contains a corresponding key: application 1 contains key K3-1, application 2 contains key K3-2, etc. These keys K3-i correspond with the keys K3-i of the set of keys K3* of Fig. 3 (i denoting the i^{th} application). As set out above, the card is able to regenerate authentication codes (e.g. MAC1, MAC2) of a command, using one or more keys (e.g. K1, K2), and to compare the regenerated codes (e.g. MAC1, MAC2) with the received codes in order to verify the received command. That is, if the received and the regenerated authentication codes (e.g. MAC2 and MAC2') do not match, the corresponding command is not loaded into the card, or at least is barred from execution. Similarly, the keys of the set K3* may be used to selectively load commands into individual applications, the target application (e.g. AP2) being indicated by the corresponding key (e.g. K3-2). In case a certain command must be loaded into more than one application, the key K3 may contain a wildcard.

As is illustrated in Fig. 4, the master file MF contains a flag register FR. In fig. 5, the flag register FR is rendered in more detail. The flag register FR comprises a plurality of flags F-1, F-2, ... of e.g. one bit each. Each flag corresponds with a command of the processor (11 in Fig. 2) of the card. The processor and/or its software is arranged in such a way that the execution of a command is inhibited if its corresponding flag is set. The UPDATE command, which updates a memory location, may e.g. only be executed if F-3 is set. Thus an additional protection is provided against the unauthorized or inadvertent execution of (application specific) commands.

The flag register can be controlled by a suitable command, e.g. SET_FLAG-i, where i is the flag number. If the commands concerned are application-specific commands, all flags are preferably initially set, thus preventing the execution of application-specific commands. A flag may be reset by a command-validating command (e.g. VALIDATE), i.e. if the authenticity of the command has been verified and the execution of the command is allowed. It will be understood that the flag register FR may be located in other files that the master file MF, and that more than one flag register may be contained in a card.

The flow diagram of Fig. 6 schematically shows an embodiment of the method of the present invention. In step 100, the procedure is initialised. This may involve producing one or more commands and offering the command to the application provider AP, who may in turn transfer the command to the card issuer CI and the third party TTP.

In step 101, the card issuer CI produces a first authentication code MAC1 of the command (COM in Fig. 3) using the first key K1. The code MAC1 is transferred to the application provider AP. The command may have been transferred to the card issuer CI in step 101 or previously, e.g. in step 100.

Similarly, in step 102, the third party TTP produces a second authentication code MAC2 of the command (COM in Fig. 3) using the second key K2. The code MAC2 is transferred to the application provider AP. The command may have been transferred to the third party in step 102 or previously, e.g. in step 100.

In step 103, the application provider AP transfers the command COM with the associated authentication codes MAC1 and MAC2 to the smart card SC (cf. Fig. 3). In step 104, the smart card SC essentially reproduces the codes MAC1 and MAC2 by producing authentication codes MAC1' and MAC2' of the command COM, using the keys K1 and K2 respectively. In step 105, the reproduced code MAC1' is compared with the received code MAC1. If the codes are equal, control is transferred to step 106, otherwise the procedure is

exited.

If the procedure is exited, the command COM received by the card is effectively inhibited, either by erasing the command or by setting a flag in the flag register. A retransmission of the command COM and its associated authentication codes may be requested. Preferably the number of retransmissions is monitored and the retransmitting may be terminated if the number of attempts exceeds a predetermined number, e.g. three or five.

In step 106, the reproduced code MAC2' is compared with the received code MAC2. If the codes are equal, control is transferred to step 107, otherwise the procedure is exited.

In step 107, the smart card SC enables the command COM, e.g. by resetting the corresponding flag in the flag register FR (cf. Fig. 5). The command COM may now be invoked and executed. In step 108, the procedure is terminated.

In the diagram of Fig. 6, only the essential steps of a preferred embodiment have been shown. Additional steps, such as the determining and evaluating of the third and fourth authentication codes MAC3 and MAC4, have been omitted for the sake of clarity. It will thus be understood by those skilled in the art that the embodiments described above are given by way of example only and that many modifications and additions are possible without departing from the scope of the present invention.

Claims

1. Method of securely loading commands (COM) in a smart card (SC) by a first party (AP), the card (SC) being issued by a second party (CI), the method comprising the steps of:

    - the second party (CI) producing a first authentication code (MAC1) of a command using a first key (K1),
    - a third party (TTP) producing a second authentication code (MAC2) of the command using a second key (K2),
    - transferring the command (COM) with the codes (MAC1, MAC2) to the card,
    - the card (SC) validating the command (COM) by reproducing the first (MAC1) and second (MAC2) authentication codes using the first (K1) and the second (K2) key respectively and comparing the reproduced codes (MAC1', MAC2') with the transferred codes (MAC1, MAC2).

2. Method according to claim 1, wherein the transferring to the card (SC) and the subsequent validation are repeated when the validation fails, the use of the loaded command (COM) being blocked until the validation succeeds.

3. Method according to claim 2, wherein the loaded command (COM) is permanently disabled if the validation fails a predetermined number (N) of times, the number (N) preferably being less than ten.

4. Method according to claim 3, wherein the disabling is performed by the card (1) changing the key (K1) of the second party (CI) stored in the card.

5. Method according to any of the preceding claims, wherein the command (COM) is an application-specific command (ASC).

6. Method according to any of the preceding claims, wherein the first (MAC1) and/or second (MAC2) authentication codes are message authentication codes produced according to the ANSI X9.19 standard.

7. Method according to any of the preceding claims, wherein the card contains several applications (e.g. AP1, AP2), each application being provided with an individual third key (e.g. K3-2) for validating a command provided with a third authentication code (MAC3), said third authentication code being produced using the individual third key (K3-2).

8. Method according to any of the preceding claims, wherein the second party (CI) produces a fourth authentication code (MAC4) of the command (COM), said fourth code (MAC4) not involving the first (K1) or second (K2) key.

9. Method according to any of the preceding claims, wherein the card comprises a flag register (FR), each flag (F-1, F-2, ...) corresponding with a command of the processor (11), the execution of a command being inhibited if its flag (e.g. F-2) is set.

10. Card (1), comprising a substrate (2) and an integrated circuit (10) having a processor (11) and a memory (12), the memory containing keys (e.g. K1, K2), <u>characterised in that</u> the integrated circuit (10) is arranged to regenerate, using at least two keys (e.g. K1, K2), authentication codes (MAC1, MAC2) of a received command (COM) and to compare regenerated authentication codes (MAC1', MAC2') with received authentication codes (MAC1, MAC2).

11. Card according to claim 10, the memory (12) comprising a data structure having individual applications (e.g. AP1, AP2), each application containing an individual key (e.g. K3-1, K3-2) for regenerating authentication codes and selectively loading commands associated with the authentication codes.

12. Card according to claim 10 or 11, the memory (12) comprising a flag register (FR), each flag (F-1, F-2, ...) corresponding with a command of the processor

(11), the execution of a command being inhibited if its flag (e.g. F-2) is set.

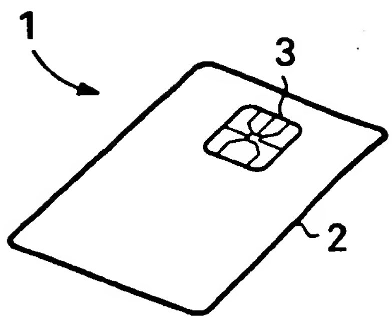

Fig. 1

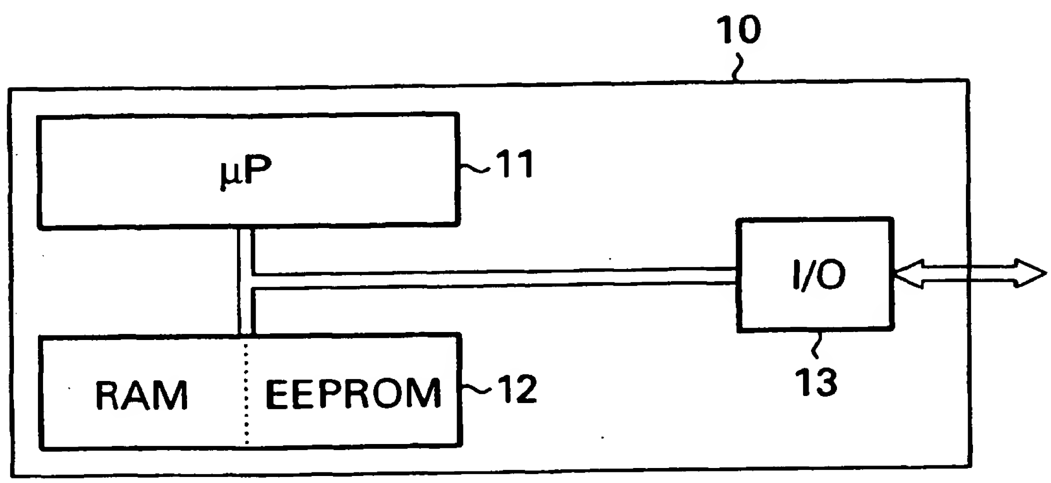

Fig. 2

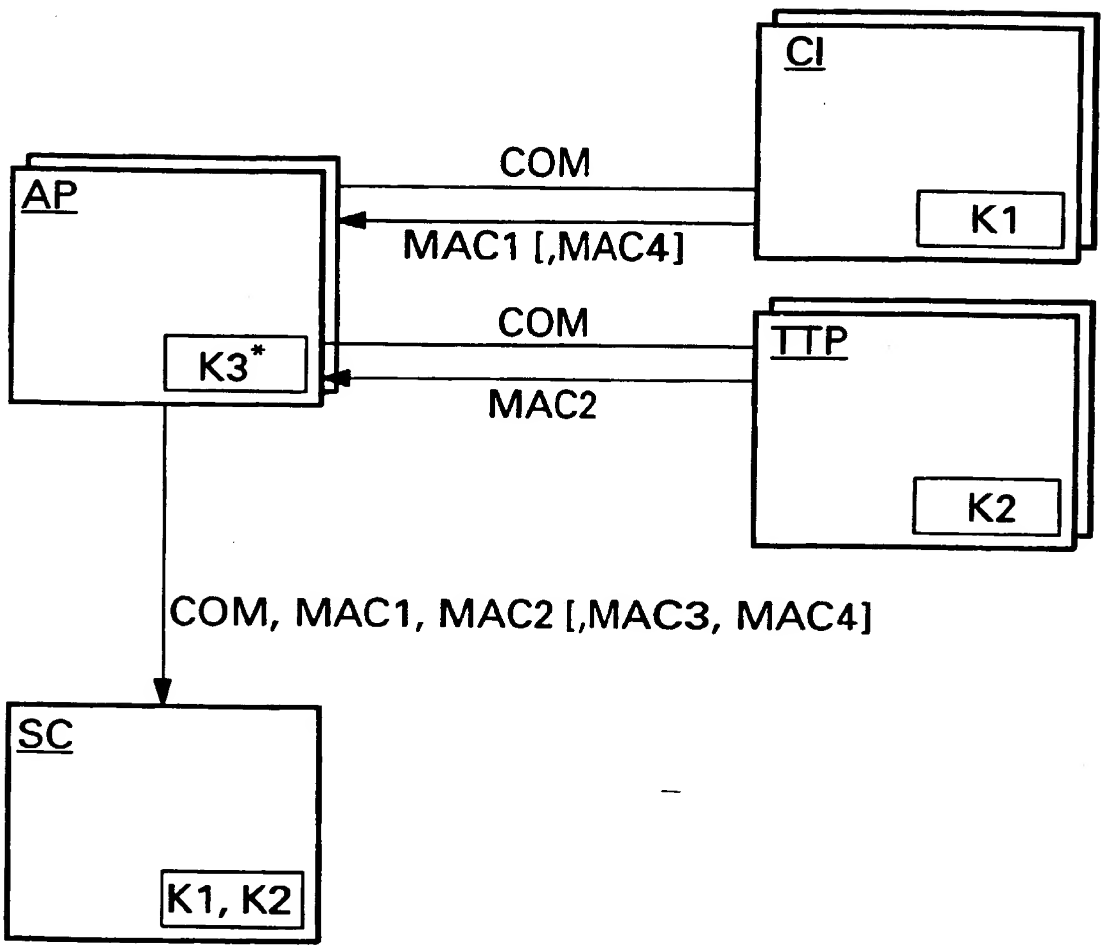
CI
K1
COM
MAC1 [,MAC4]
AP
K3*
COM
MAC2
TTP
K2
COM, MAC1, MAC2 [,MAC3, MAC4]
SC
K1, K2

Fig. 3

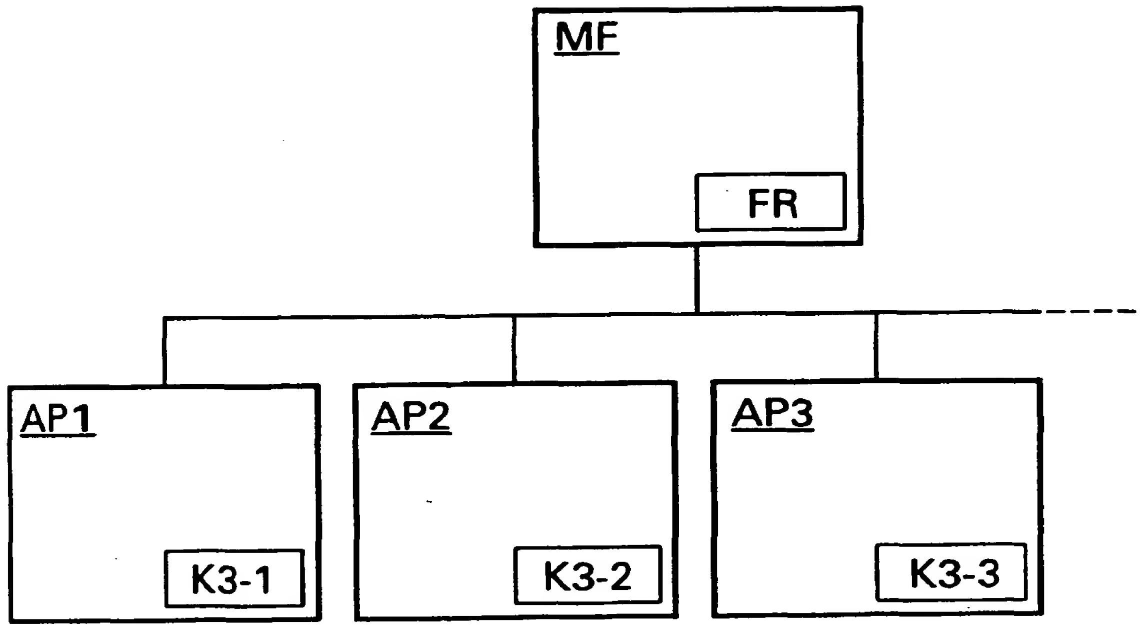

Fig. 4

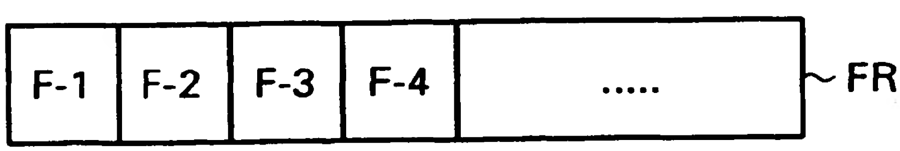

Fig. 5

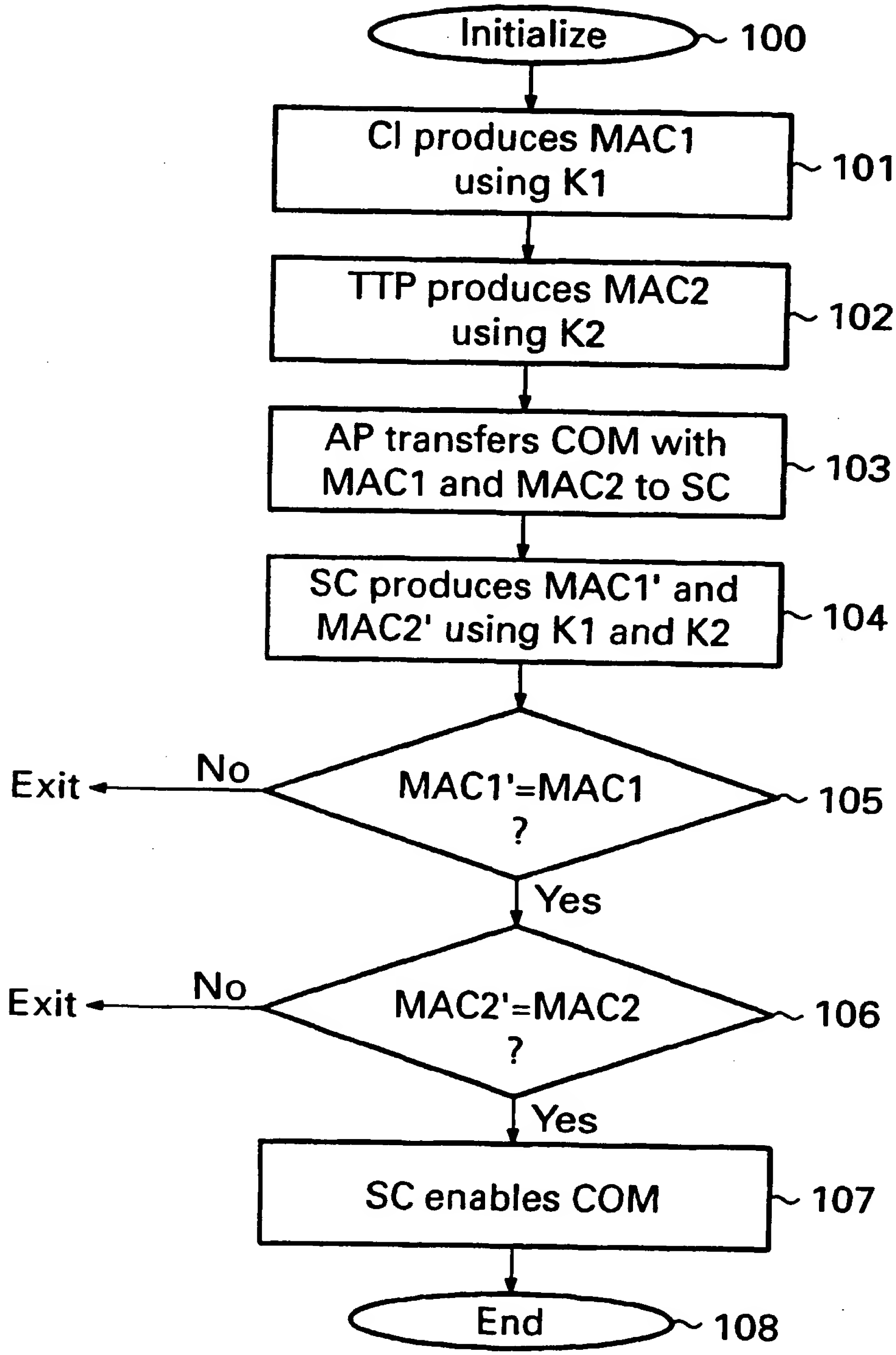
Initialize
100
CI produces MAC1 using K1
101
TTP produces MAC2 using K2
102
AP transfers COM with MAC1 and MAC2 to SC
103
SC produces MAC1' and MAC2' using K1 and K2
104
MAC1'=MAC1 ?
105
No
Exit
Yes
MAC2'=MAC2 ?
106
No
Exit
Yes
SC enables COM
107
End
108

Fig. 6

European Patent Office

EUROPEAN SEARCH REPORT

Application Number
EP 96 20 0867

DOCUMENTS CONSIDERED TO BE RELEVANT

Category	Citation of document with indication, where appropriate, of relevant passages	Relevant to claim	CLASSIFICATION OF THE APPLICATION (Int.Cl.6)
A	EP-A-0 218 176 (TOSHIBA) * abstract; claims; figures 1-9 * * column 3, line 25 - column 6, line 30 * ---	1,5,9,10	G07F7/10
A	DE-A-41 19 924 (SIEMENS) * abstract; claims; figures 1-3 * * column 1, line 1 - column 3, line 64 * -----	1,6,10	
			TECHNICAL FIELDS SEARCHED (Int.Cl.6)
			G07F

The present search report has been drawn up for all claims

Place of search	Date of completion of the search	Examiner
THE HAGUE	14 October 1996	David, J

CATEGORY OF CITED DOCUMENTS

X : particularly relevant if taken alone
Y : particularly relevant if combined with another document of the same category
A : technological background
O : non-written disclosure
P : intermediate document

T : theory or principle underlying the invention
E : earlier patent document, but published on, or after the filing date
D : document cited in the application
L : document cited for other reasons

& : member of the same patent family, corresponding document

EPO FORM 1503 03.82 (P04C01)

This Page Blank (uspto)